

Lucas Burgardt

13/11/2024

<https://github.com/Niip07/QA-Challenge>

1. Introdução

Este documento detalha o plano de testes para a API de piadas que fornece piadas aleatórias, conforme o endpoint base https://official-joke-api.appspot.com/random_joke. O objetivo é garantir a funcionalidade, performance e a robustez da API.

2. Objetivos dos Testes

- Verificar se a API retorna respostas corretas no formato esperado.
- Garantir a integridade e a unicidade dos dados retornados.
- Avaliar a performance da API sob carga.
- Documentar possíveis falhas e sugerir melhorias.

3. Escopo dos Testes

Os testes abrangem:

- Validação de resposta (formato e tipos de dados).
- Testes de unicidade dos IDs.
- Testes de performance e carga com múltiplos usuários simultâneos.
- Verificação de consistência em diferentes chamadas.

4. Critérios de Aceitação

- Todas as respostas devem estar no formato JSON e conter os campos `type`, `setup`, `punchline` e `id`.
- O campo `id` deve ser único em cada chamada.
- Nenhum campo deve estar vazio.
- O tempo de resposta deve ser aceitável (≤ 500 ms em média).

5. Casos de Teste

ID	Descrição	Passos	Resultado Esperado
TC001	Verificar formato da resposta	Fazer uma requisição e verificar se a resposta está em formato JSON	A resposta deve estar em formato JSON
TC002	Validar presença de todos os campos	Checar se <code>type</code> , <code>setup</code> , <code>punchline</code> e <code>id</code> estão presentes	Todos os campos devem estar presentes
TC003	Verificar tipos de dados	Validar que <code>id</code> é numérico e <code>setup/punchline</code> são strings	Os tipos de dados devem corresponder
TC004	Verificar unicidade do ID	Fazer 100 requisições e checar se todos os IDs são diferentes	IDs devem ser únicos em todas as respostas
TC005	Testar performance (tempo de resposta)	Simular 10 usuários fazendo requisições simultâneas	Tempo de resposta ≤ 500 ms
TC006	Verificar consistência de resposta	Fazer chamadas consecutivas e comparar a integridade das respostas	Respostas devem ser consistentes e sem campos faltando

6. Ferramentas Utilizadas

- **Automação:** Cypress
- **Documentação de Resultados:** Planilhas e relatórios gerados por ferramentas de análise.

7. Riscos e Dependências

- **Dependências de Rede:** A execução dos testes depende de uma conexão de internet estável e confiável para evitar falhas durante as requisições.
- **Limitações de Taxa da API:** A API pode ter limitações de taxa que impactam os testes de carga e a frequência de requisições. Isso pode resultar em respostas com status de erro (e.g., `429 Too Many Requests`).
- **Disponibilidade da API:** Se a API estiver indisponível devido a manutenção ou problemas técnicos, isso pode atrasar a execução dos testes e a coleta de resultados.
- **Ambiente de Teste:** O ambiente onde os testes são executados (máquina local, servidor, etc.) pode influenciar a latência das requisições e as medições de tempo de resposta.
- **Limitações de Ferramentas:** Ferramentas de automação e carga, como JMeter e locust.io, podem ter suas próprias limitações em termos de escalabilidade e capacidade de simulação.

- **Erros de Dados:** Respostas com dados inconsistentes ou que não seguem o formato documentado podem gerar falsos negativos durante a execução dos testes.
- **Configuração de Usuários Simultâneos:** A simulação de múltiplos usuários simultâneos pode requerer ajustes na configuração da máquina de teste para garantir a precisão e a performance dos testes de carga.
- **Fatores Externos:** Alterações na API por parte da equipe de desenvolvimento sem aviso prévio podem impactar os resultados e exigem revisões nos testes planejados.